

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
УКРАЇНИ “КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ
ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО”
ФАКУЛЬТЕТ ІНФОРМАТИКИ ТА ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ

Кафедра автоматики та управління в технічних системах

«До захисту допущено»

Завідувач кафедри

(підпис)

(ініціали, прізвище)

“ ” _____ 2019 р.

Дипломний проект

на здобуття ступеня бакалавра

з напряму підготовки

6.050103 «Програмна інженерія»

спеціальність

«Програмна інженерія»

на тему: Система розпізнавання архітектурних елементів на основі
нейронних мереж

Виконав:

студент 4 курсу, групи ІТ-51

Клапішевський Андрій Сергійович

(прізвище, ім'я, по батькові)

(підпис)

Керівник

доцент с.н.с, к.т.н. Савчук О. В.

(посада, вчене звання, науковий ступінь, прізвище та ініціали)

(підпис)

**Консультант з
графічної**

документації

доцент с.н.с, к.т.н. Савчук О. В.

(посада, вчене звання, науковий ступінь, прізвище та ініціали)

(підпис)

Рецензент

(посада, вчене звання, науковий ступінь, прізвище та ініціали)

(підпис)

Засвідчую, що у цьому дипломному

проекті немає запозичень з праць інших
авторів без відповідних посилань.

Студент _____

(підпис)

Київ – 2019 року

Національний технічний університет України
“Київський політехнічний інститут імені Ігоря Сікорського”

Факультет (інститут) _____ Інформатики та обчислювальної техніки

(повна назва)

Кафедра _____ автоматики та управління в технічних системах

(повна назва)

Рівень вищої освіти – перший (бакалаврський)

Напрямок підготовки (програма професійного спрямування) – 6.050103

«Програмна інженерія»

ЗАТВЕРДЖУЮ

Завідувач кафедри

(підпис)

(ініціали, прізвище)

“ ”

ЗАВДАННЯ

НА ДИПЛОМНИЙ ПРОЕКТ СТУДЕНТУ

Клапішевському Андрію Сергійовичу

(прізвище, ім'я, по батькові)

1. Тема проекту «Система розпізнавання архітектурних елементів на основі нейронних мереж»

керівник проекту Савчук О. В *с.н.с, к.т.н.*

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджена наказом по університету від « _____ » _____ № _____

2. Термін подання студентом проекту « _____ » _____

3. Вихідні дані до проекту

Технічне завдання

4. Зміст пояснювальної записки

- 1) Аналіз вимог до програмного забезпечення: основні визначення та терміни, опис предметного середовища, огляд існуючих технічних рішень та відомих програмних продуктів, розробка функціональних та нефункціональних вимог
- 2) Моделювання програмного забезпечення: моделювання та аналіз програмного забезпечення, засоби розробки, технічні рішення, архітектура програмного забезпечення
- 3) Розгортання та впровадження програмного забезпечення
- 4) Керівництво користувача, методика та програма тестування

5. Перелік графічного матеріалу

- 1) Структурна схема архітектури мережі. Нова модель
- 2) Схема алгоритму навчання мережі на графічних прискорювачах
- 3) Схема алгоритму навчання мережі методом зворотнього поширення помилок
- 4) Структурна схема взаємозв'язків програмного забезпечення для реалізації глибокого навчання нейронних мереж

6. Консультанти розділів проекту

| | Прізвище, ініціали та посада | Підпис, дата | |
|--|------------------------------|--------------|--|
| | | | |
| | | | |

7. Дата видачі завдання « 13 » березня 2019 р.

КАЛЕНДАРНИЙ ПЛАН

| № | Назва етапів виконання дипломного | Термін виконання | Примітка |
|---|-------------------------------------|------------------|----------|
| 1 | Вивчення предметної області | 27.02.2019 | |
| | Аналіз існуючих методів розв'язання | | |
| 3 | Постановка та формалізація задачі | 18.03.2019 | |
| | Аналіз вимог до програмного | | |
| | Моделювання програмного | | |
| | Обґрунтування використовуваних | | |
| | Розробка архітектури програмного | | |

| | | | |
|----|-----------------------------------|------------|--|
| 8 | Розробка програмного забезпечення | 05.05.2019 | |
| 9 | Налагодження програми | 10.05.2019 | |
| 10 | Виконання графічних документів | 16.05.2019 | |
| 11 | Оформлення пояснювальної записки | 20.05.2019 | |
| 12 | Подання ДП на попередній захист | 04.06.2019 | |
| 13 | Подання ДП рецензенту | 13.06.2019 | |
| 14 | Подання ДП на основний захист | 18.06.2019 | |

Студент

Клапішевський А.С.

(підпис)

Керівник проекту

Савчук О.В.

(підпис)

| № з/п | Формат | Позначення | Найменування | лістів Кількість | Примітка |
|-------|--------|--------------------|--|---------------------|----------|
| 1 | A4 | IT51.090БАК.005 ПЗ | Завдання на дипломний проект | 1 | |
| 2 | A4 | | | | |
| 3 | A3 | IT51.090БАК.005 ПЗ | Пояснювальна записка | 55 | |
| 4 | A3 | IT51.090БАК.005 СС | Схема структурна функціональних вимог | 1 | |
| 5 | A3 | IT51.090БАК.005 СС | Схема структурна бізнес-процесів застосунку | 1 | |
| | | IT51.090БАК.005 СС | Схема структурна класів програмного забезпечення | 1 | |
| | | IT51.090БАК.005 СС | | | |

| | | | | | | | | |
|----------|---------------------|-----|-------|------|-----------------|------|------|--------|
| | | | | | IT51.090БАК.005 | | | |
| | | | | | | | | |
| Зм. | Арк. | ПІБ | Підп. | Дата | | | | |
| Розробн. | Клапішевський А.С.. | | | | Відомість | Лім. | Лист | Листів |
| Керівн. | | | | | | | 1 | 1 |
| Консульт | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |

АНОТАЦІЯ

Пояснювальна записка дипломного проекту складається з 6 розділів, містить 12 рисунків, 4 таблиць, 12 посилань – загалом 58 сторінки.

Об'єкт дослідження: система, що може викристовуватися на підприємстві для більш точної оцінки вартості нерухомості.

Мета дипломного проекту: полегшити життя як звичайної людини, так і працівника ріелторської фірми.

У першому розділі описано задачу та теоретичні основи навчання, будови та проектування нейронних мереж.

У другому розділі порівнюємо різні методики будови системи нейронних мереж та виокремлюємо найраціональнішу модель.

У третьому розділі проведено аналіз засобів реалізації, та виокремлення стеку технологій, що максимально задовольняють всі вимоги до реалізації.

У четвертому розділі описано розробку застосунку.

У п'ятому розділі представлені результати експериментів з нейронними мережами та набором даних.

У додатках наведено: схеми методів навчання нейронних мереж, графічне представлення результатів експериментів,.

ABSTRACT

The explanatory note of the diploma project consists of 6 sections, containing 12 figures, 4 tables, 12 references - a total of 58 pages.

Object of research: a system that can be used by the enterprise to more accurately estimate the value of real estate.

The purpose of the diploma project: to facilitate life as an ordinary person and an employee of the real estate firm.

The first chapter describes the problem and the theoretical foundations of learning, structure and design of neural networks.

In the second section, we compare the different methods of building the system of neural networks and highlight the most rational model.

In the third section, the analysis of the means of implementation, and the isolation of the stack of technologies that maximally meet all the requirements for implementation, is carried out.

The fourth section describes the development of the application.

The fifth section presents the results of experiments with neural networks and a set of data.

The annexes show: diagrams of neural network teaching methods, graphical presentation of results of experiments.

**Пояснювальна записка
до дипломного проекту
на тему: «Система розпізнавання архітектурних
елементів на основі нейронних мереж»**

Київ – 2019 рік

ЗМІСТ

| | |
|---|----|
| Зміст..... | 5 |
| ПЕРЕЛІК СКОРОЧЕНЬ..... | 7 |
| Вступ..... | 3 |
| 1. Постановка задачі | 4 |
| 2. Теоретичні основи машинного та глибокого навчання | 5 |
| 2.1 Основи машинного навчання | 5 |
| 2.2 Основи глибокого навчання | 9 |
| 2.3 Порівняння методів машинного та глибокого навчання..... | 11 |
| 2.4 Порівняння методів машинного та глибокого навчання..... | 12 |
| Висновки до розділу 2 | 13 |
| 3. аналіз ЗАСОБІВ РЕАЛІЗАЦІЇ | 14 |
| 3.1 Обґрунтування вибору типу моделі..... | 14 |
| 3.2 Опис набору даних | 15 |
| 3.3 Фільтр анізотропної функції..... | 16 |
| Висновки до розділу 3 | 18 |
| 4. ОСНОВНА СИСТЕМА | 19 |
| 4.1 Опис основної системи | 19 |
| 4.2 Підтримка векторної регресії (SVR)..... | 21 |
| 4.3 Оцінка ефективності..... | 22 |
| Висновки до розділу 4 | 23 |
| 5. ЕКСПЕРЕМЕНТИ ТА РЕЗУЛЬТАТИ..... | 24 |
| 5.1 SVR експерименти | 24 |
| 5.2 Експеримент з нейронною мережею | 24 |
| 5.3 Нейронна мережа з набору даних будинків..... | 25 |

| | | | | | | | | |
|-----------|---------------|----------|--------|------|---|---|-------|---------|
| | | | | | IT51.200БАК.009 ПЗ | | | |
| Зм. | Арк. | № докум. | Підпис | Дата | Система розпізнавання архітектурних елементів на основі нейронних мереж Пояснювальна записка | Літ. | Аркуш | Аркушів |
| Розробив | Клапішевський | | | | | | 2 | 116 |
| Перевірив | Тимофєєва Ю.С | | | | | | | |
| Реценз. | | | | | | | | |
| Н. Контр. | Шинкевич М.К | | | | | | | |
| Затвердив | | | | | | КПІ ім. Ігоря Сікорського, ФІОТ група IT-51 | | |

| | | |
|-------|--|---------------------------------|
| 1.1.1 | Слой активации | 26 |
| 1.1.2 | Слой пулинга..... | Ошибка! Закладка не определена. |
| 1.1.3 | Слой нормализации | Ошибка! Закладка не определена. |
| 1.1.4 | Полносвязный слой | Ошибка! Закладка не определена. |
| 1.1.5 | Softmax-слой..... | Ошибка! Закладка не определена. |
| 1.2 | Оптимизация производительности | Ошибка! Закладка не определена. |
| 1.2.1 | Наборы данных..... | Ошибка! Закладка не определена. |
| 1.2.2 | Момент | Ошибка! Закладка не определена. |
| 1.2.3 | Кросс-валидация | Ошибка! Закладка не определена. |
| 1.2.4 | Инициализация весов .. | Ошибка! Закладка не определена. |
| 1.2.5 | Дропаут | Ошибка! Закладка не определена. |
| 1.2.6 | L2-регуляризация | Ошибка! Закладка не определена. |
| 1.2.7 | Ранняя остановка | Ошибка! Закладка не определена. |
| 1.2.8 | Аугментация данных .. | Ошибка! Закладка не определена. |
| 1.3 | Количество сверточных слоев и карт признаков | Ошибка! Закладка не определена. |
| 1.4 | Методология | Ошибка! Закладка не определена. |
| 1.4.1 | Распознавание регистрационного номера | Ошибка! Закладка не определена. |

Список використаної літератури 37

ДОДАТОК Г

ДОДАТОК Д

| | | | | | | | | |
|-----------|---------------|----------|--------|------|---|---|-------|---------|
| | | | | | IT51.200БАК.009 ПЗ | | | |
| Зм. | Арк. | № докум. | Підпис | Дата | Система розпізнавання архітектурних елементів на основі нейронних мереж Пояснювальна записка | Літ. | Аркуш | Аркушів |
| Розробив | Клапішевський | | | | | | 2 | 116 |
| Перевірив | Тимофєєва Ю.С | | | | | | | |
| Реценз. | | | | | | | | |
| Н. Контр. | Шинкевич М.К | | | | | | | |
| Затвердив | | | | | | КПІ ім. Ігоря Сікорського, ФІОТ група IT-51 | | |

ПЕРЕЛІК СКОРОЧЕНЬ

Графічні прискорювачі (англ. graphic processing unit, GPU) – окремий пристрій персонального комп'ютера або ігрової приставки, що виконує графічний рендеринг.

ЗНМ – згорткова нейронна мережа.

Кластер (англ. cluster) – група однакових або подібних елементів, зібраних разом або близько розташованих один до одного.

Набір даних (англ. data set) – колекція однотипних даних, що застосовується в задачах машинної обробки даних.

Синапс нейрона (від грецького «synapsis» – з'єднання) – структура, яка дозволяє нейрону (нейронної мережі) проводити сигнал (у випадку штучного нейрону) до іншого нейрону.

Тензор (від лат. tendere, «тягнутись, простиратися») – математичний об'єкт, що узагальнює такі поняття як скаляр, вектор, ковектор, лінійний оператор і білінійна форма. Вивченням тензорів займається тензорне числення.

BN – Batch-нормалізація.

| | | | | | | | | |
|-----------|---------------|----------|--------|------|---|---|-------|---------|
| | | | | | IT51.200БАК.009 ПЗ | | | |
| Зм. | Арк. | № докум. | Підпис | Дата | Система розпізнавання архітектурних елементів на основі нейронних мереж Пояснювальна записка | Літ. | Аркуш | Аркушів |
| Розробив | Клапішевський | | | | | | 2 | 116 |
| Перевірив | Тимофєєва Ю.С | | | | | | | |
| Реценз. | | | | | | | | |
| Н. Контр. | Шинкевич М.К | | | | | | | |
| Затвердив | | | | | | КПІ ім. Ігоря Сікорського, ФІОТ група IT-51 | | |

ВСТУП

Сучасний світ стрімкими кроками прямує в еру повної автоматизації процесів. Обчислювальні машини, роботизовані організми та нейронні мережі займають провідне місце у сфері сучасної кібернетики, поступово заміщуючи людський фактор на ринку праці. Концепції «розумного будинку» і «розумного міста», розвиток автопілотування і систем штучного інтелекту – ось перелік незначної частини речей, в яких людство досягло неймовірного прогресу, що дозволило не тільки підвищити рівень комфорту кожного представника прогресивного суспільства, а й покращити показники точності, якості та швидкості виконання поставлених задач.

В рамках свого дипломного проекту програмний продукт на базі нейронної мережі. Мій вибір ґрунтується на показниках, що засвідчують повне домінування даної технології над іншими системами в області розпізнавання елементів зображення. Ефективність глибокої згорткової мережі полягає в гібридному підході до архітектурних рішень та додаткових методів, що запобігають перенавчанню системи.

В даній роботі представлено архітектуру глибокої згорткової мережі, що реалізує нейромережеву систему для розпізнавання архітектурних об'єктів на зображеннях, застосовуючи власний підхід до класифікації з використанням ієрархічного класифікатора.

Об'єктом дослідження даної роботи виступають завдання аналізу, перетворення і розпізнавання цифрових зображень, предмет позначений структурою самого дослідження. Метою роботи є дослідження цих підходів і реалізація системи розпізнавання елементів архітектури на зображеннях. Висока точність розпізнавання, що дозволяє рекомендувати систему для використання в комерційних цілях, поряд з використанням сучасних технологій становить практичну значимість роботи.

| | | | | | | |
|------|------|----------|--------|------|--------------------|------|
| | | | | | IT51.200БАК.007 Д5 | Лист |
| Ізм. | Лист | № докум. | Підпис | Дата | | |

1. ПОСТАНОВКА ЗАДАЧІ

Мета роботи – побудувати модель згорткової нейронної мережі, призначеної для задачі розпізнавання архітектурних елементів на зображеннях та формування довідкової інформації на основі статистичних даних .

Для досягнення мети було виокремлено наступні задачі:

- отримати основні теоретичні відомості про машинне та глибоке навчання;
- виконати дослідження основних методів попередньої обробки, сегментації та розпізнавання елементів архітектури на зображеннях;
- запропонувати методи для сегментації і розпізнавання символів на зображеннях;
- побудувати модель нейронної мережі, призначеної для задачі розпізнавання архітектурних елементів на зображеннях;
- виконати програмну реалізацію системи розпізнавання елементів архітектури з використанням згорткової нейронної мережі.

Наукова новизна роботи полягає у високій надійності і точності запропонованого метода, комплексному розгляді теоретичних і практичних нюансів, що лежать в його основі алгоритмів, а також в обґрунтуванні їх застосування в задачі розпізнавання. Також, розроблена система стане фундаментом для її подальшого вдосконалення і розвитку в повноцінний програмний продукт, що полегшить життя звичайних людей шляхом заміщення послуг ріелторських фірм. Дане рішення дозволить повністю виключити людський фактор з процедури оцінювання вартості житлової площі та нерухомості загалом, що, в свою чергу, підвищить точність та об'єктивність результатів.

| | | | | | | |
|------|------|----------|--------|------|--------------------|------|
| | | | | | IT51.200БАК.007 Д5 | Лист |
| Ізм. | Лист | № докум. | Підпис | Дата | | |

2. ТЕОРЕТИЧНІ ОСНОВИ МАШИННОГО ТА ГЛИБОКОГО НАВЧАННЯ

2.1 Основи машинного навчання

Розпізнавання образів полягає в пошуку та ідентифікації об'єктів на зображенні, причому ідентифікація представляє собою відношення кожного з вихідних об'єктів (вихідної інформації) до одного або декількох класів шляхом виділення у цих об'єктів певних ознак (до них можуть відноситись, наприклад, межі, кути, відмінності кольорової схеми і т.д.). Для цих цілей теорія розпізнавання образів використовує методи машинного навчання, які стають все більш практичними і популярними протягом багатьох років, аж до їх повсякденного використання.

Для вирішення завдання класифікації тексту, тобто встановлення зв'язку між символами тексту і класами, до яких вони належать, може використовуватися алгоритм контрольованого навчання, або навчання з учителем, який полягає в демонстрації системі розпізнавання декількох навчальних зображень, кожному з яких поставлена у відповідність мітка класу (відгук), зроблена, можливо, людиною (який і є вчителем). Сукупність таких пар «об'єкт - відповідь» при цьому називається навчальною вибіркою. Набір пар, які системі розпізнавання невідомі, використовується для оцінки якості алгоритму і називається тестовою вибіркою.

Двома найважливішими типами задач навчання з учителем є регресія і класифікація. У задачі класифікації алгоритм намагається передбачити відгук для невивчених даних на основі даних навчання, і безліч можливих відповідей, таким чином, є кінцевим. У задачі регресії вихід, який намагається передбачити алгоритм, є безперервним, тобто відповіді є дійсними числами або векторами дійсних чисел.

Структурна схема на малюнку 2.1 ілюструє процес навчання з учителем [5]. Тут участь вчителя розглядається як наявність знань про навколишнє середовище, причому ці знання представлені парами «вхід-вихід». При

| | | | | | | |
|------|------|----------|--------|------|--------------------|------|
| | | | | | ІТ51.200БАК.007 Д5 | Лист |
| Ізм. | Лист | № докум. | Підпис | Дата | | |

цьому саме навколишнє середовище невідоме системі, що навчається. Якщо на вхід вчителю і системі, що навчається, подається навчальний вектор з навколишнього середовища, то вчитель на основі наявних знань може сформувати і передати системі бажаний відгук (дії, які повинна виконати система), який відповідає цьому вхідному вектору. Різниця між бажаним і поточним відгуком системи називається сигналом помилки. Параметри системи налаштовуються за допомогою комбінації навчальної вибірки і сигналу помилки з використанням різних методів оптимізації; налаштування відбувається покроково для того, щоб система імітувала поведінку вчителя. Оптимальні параметри дозволяють системі коректно визначати відгук для раніше невидимих векторів, оскільки знання вчителя передаються в систему в максимально повному обсязі.

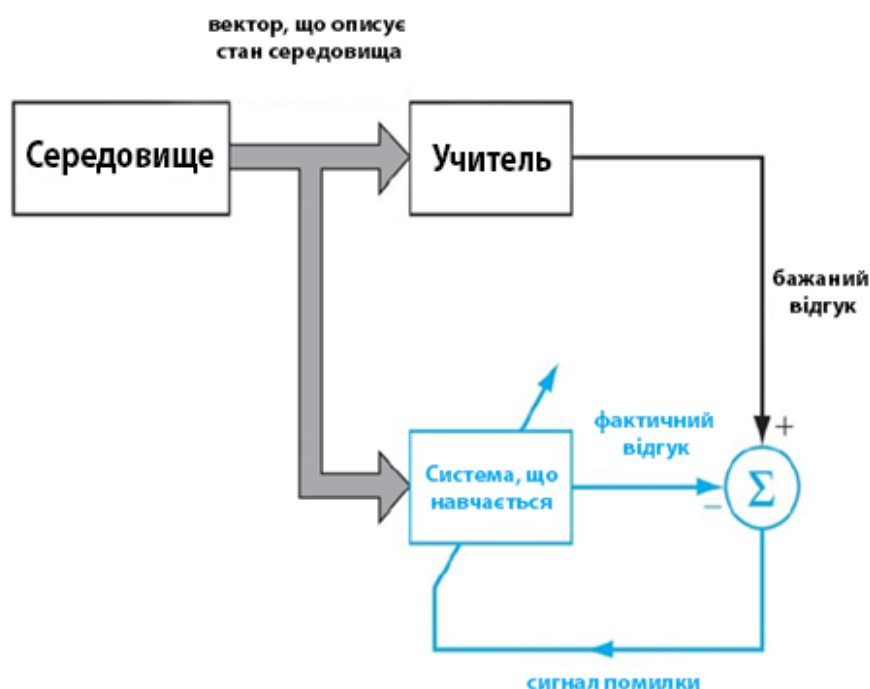


Рисунок 2.1 – Структурна схема навчання з учителем

Іншим великим розділом машинного навчання є навчання без вчителя (рисунок 2.2). У цьому виді навчання системі надаються дані, а бажаний

відгук не передбачений. Класичний приклад використання такого підходу - завдання кластеризації, де задана множина об'єктів розбивається на підмножини (кластери). Об'єкти виділяються в підмножини завдяки спільним властивостям, в результаті чого кожна з підмножин складається зі схожих об'єктів, а між об'єктами різних підмножин існують серйозні відмінності. Зовсім нещодавно, особливо з появою технологій глибокого навчання, навчання без вчителя стало корисним інструментом попередньої обробки в задачах виявлення корисних уявлень даних [6].



Рисунок 2.2 – Структурна схема навчання без учителя

Для безпосереднього процесу навчання використання повнорозмірних вхідних даних може бути невиправданим або зовсім неможливим. З цієї причини дані, як правило, піддаються невеликій попередній обробці, в результаті якої алгоритми машинного навчання в якості вхідних даних використовують найбільш важливі ознаки, витягнуті з вихідних даних. Проблема такого підходу полягає в пошуку та виборі тих ознак, які є більш важливими, ніж інші.

Однією з ключових завдань у машинному навчанні є здатність алгоритму до узагальнення. Висновок закономірностей з прецедентів означає можливість добре працювати на нових, раніше невивчених вхідних даних, яку повинен забезпечити алгоритм, що використовується.

Найчастіше навчальна помилка (тобто помилка, що виникає в процесі навчання) може бути зменшена. Однак основна відмінність машинного навчання від методів оптимізації полягає в тому, щоб також була низькою помилка узагальнення, названа тестовою помилкою.

Тестова помилка визначається як очікуване значення помилки на новому вході і зазвичай оцінюється шляхом вимірювання продуктивності алгоритму на тестовій вибірці прикладів, які були зібрані окремо від навчальної вибірки. Величина середньої помилки навчання і невеликий розрив між помилкою навчання і тестової помилкою є факторами, які впливають на дві центральні проблеми в машинному навчанні, - недонавчання і перенавчання. Недонавчання відбувається в тому випадку, коли алгоритм не може отримати достатньо низьке значення помилки на навчальній вибірці. Така ситуація може виникнути при використанні недостатньо складних методів. Перенавчання відбувається, коли розрив між помилкою навчання і значенням тестової помилки занадто великий. Як правило, до цього призводить складний метод навчання в поєднанні з невеликим розміром навчальної вибірки. Мета етапу навчання полягає в мінімізації помилки.

На малюнку 2.3 показані ефекти недообученого і перенавчання.

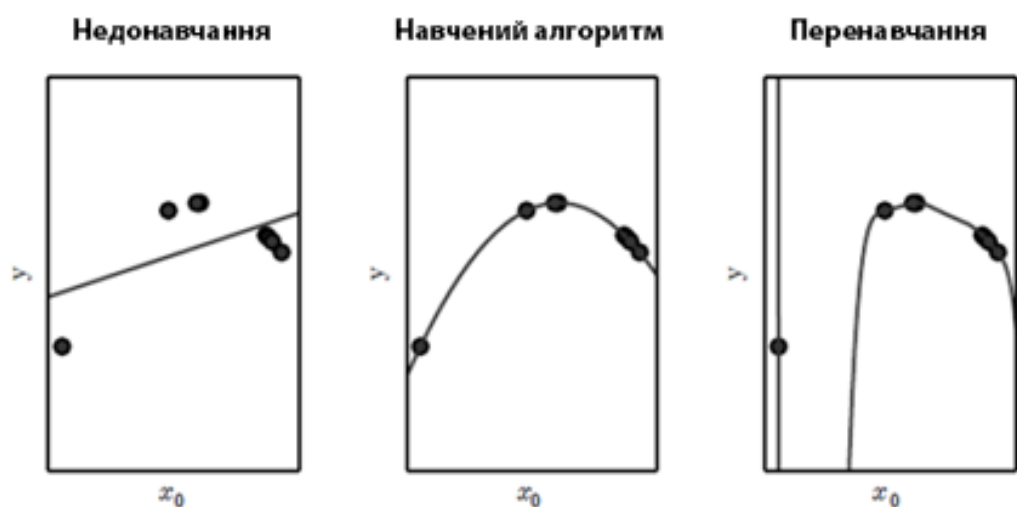


Рисунок 2.3 – Структурна схема навчання без учителя

На графіку зліва функція навчання охоплює не всі крапки (відбувається недонавчання), на графіку праворуч виникає «помилкове спрацьовування» - в підсумковому результаті включається точка, яка є шумом (перенавчання). На графіку в центрі проілюстрований приклад навченого алгоритму.

2.2 Основи глибокого навчання

Глибоке навчання - це нова навчальна парадигма, також відома як глибоке структуроване навчання, яка є активною областю дослідження нейронних мереж, штучного інтелекту, розпізнавання образів та інших дисциплін. Вона характеризується як сукупність алгоритмів машинного навчання, яка [7]:

- виконує автоматичне виділення ознак, використовуючи систему багат шарових нелінійних функціональних блоків (фільтрів). Кожен наступний шар використовує вихід одного або декількох попередніх шарів в якості вхідних даних;
- Виявляє ознаки високого рівня за допомогою ознак низького рівня. Організація ознак є ієрархічною для кожного з видів навчання (з учителем і без). Крім того, при навчанні без учителя використовуються параметри подання даних;
- Вивчає дані, моделюючи складні відносини між ними. Таким чином, глибоке навчання є частиною більш широкої області машинного навчання - вивчення уявлень даних.

Методи глибокого навчання засновані на розподіленому поданні даних. Основне припущення, що лежить в основі розподілених представлень, полягає в тому, що дані виходять в результаті взаємодії великої кількості факторів на

| | | | | | | |
|------|------|----------|--------|------|--------------------|------|
| | | | | | IT51.200БАК.007 Д5 | Лист |
| Ізм. | Лист | № докум. | Підпис | Дата | | |

різних рівнях ієрархії. Глибоке навчання передбачає, що ці чинники можуть бути об'єднані в шари, які відповідають різним рівням абстракції або композиції. Кількість шарів і їх розмір може варіюватися, тим самим забезпечуючи різні рівні абстракції. Основні причини використання глибокого навчання в поточних дослідженнях пов'язані з сучасними високоефективними графічними процесорами, досягненнями в машинному навчанні і наявністю великого обсягу даних для навчання і тестування.

Приклади останніх досліджень в цій області:

- GoogLeNet [8] - глибока мережу з 22 шарами, яка була представлена на конкурсі ILSVRC-2014 і посіла там перше місце з помилкою 6,67%. Завдання полягало в тому, щоб призначити вхідному зображенню мітку одного з тисячі класів;
- R-CNN (Region-based Convolutional Network, рус. Згорткові нейронні мережі на основі вікон кандидатів) є однією з сучасних технологій розпізнавання об'єктів, запропонованої Россом Гіршіком спільно з групою вчених з інституту Берклі [9]. R-CNN виявляє об'єкт на зображенні, виконуючи семантичну сегментацію;
- Архітектура нейронної мережі, що надається лабораторією Google X [10], складається з одного мільярда вузлів і використовує 16 000 комп'ютерних процесорів. Система домоглася точності 74,8% при розпізнаванні кішок, 76,7% при розпізнаванні людських частин тіла і 81,7% при виявленні людських облич.

Провідними дослідниками в області глибокого навчання є Google, Facebook, Університети Берклі і Торонто. Їхні сьогоднішні розробки привели до практичного успіху в різних додатках комп'ютерного зору, при розпізнаванні мови і образів, а також при використанні згорткових нейронних мереж.

| | | | | | | |
|------|------|----------|--------|------|--------------------|------|
| | | | | | IT51.200БАК.007 Д5 | Лист |
| Ізм. | Лист | № докум. | Підпис | Дата | | |

2.3 Порівняння методів машинного та глибокого навчання

Методи глибокого навчання представляють собою потужні моделі машинного навчання, які забезпечують кращу продуктивність для задач розпізнавання об'єктів на зображеннях [11] і розпізнавання мови [12]. Традиційні методи машинного навчання займаються отриманням ознак з доступних вхідних даних. Класифікатор, отримавши витягнуті раніше ознаки, призначає кожному вхідному вектору мітку відповідного класу. Виділення ознак, таким чином, є процедурою вивчення і отримання цінних даних з вхідних даних. Основні детектори ознак в комп'ютерному зорі - це SIFT (Scale-Invariant Feature Transform, рус. Масштабно-інваріантне функціональне перетворення) [13] і HOG (Histogram of Oriented Gradients, рус. Гістограма спрямованих градієнтів) [14]. Типова методика навчання вимагає ретельного ручного інжинірингу блоків виділення характерних ознак, причому пошук кожної ознаки є комплексним завданням, яка залежить від предметної області. На малюнку 2.4 проілюстрована структурна схема традиційної моделі [15].



Рисунок 2.4 – Структурна схема традиційних методів машинного навчання

Системи глибокого навчання перевершують можливості звичайних систем. Процес вибору та налаштування ознак може бути автоматизований - методи глибокого навчання ієрархічно отримують необхідні ознаки з вхідних даних і вчаться за допомогою самої навчальної вибірки без вчителя або з частковим залученням вчителя [16]. В останньому випадку

використовується невелика кількість розмічених і велика кількість нерозмічених даних. Типова модель глибокого навчання представлена на рисунку 2.5 [15].



Рисунок 2.5 – Структурна схема моделі глибокого навчання

2.4 Порівняння методів машинного та глибокого навчання

У машинному навчанні функція вартості, також відома як функція втрат [17], це показник того, наскільки результат, отриманий від системи машинного навчання, узгоджується з бажаним класом даних навчання. Метою навчання з учителем є мінімізація втрат, тобто оптимальне рішення представлене у вигляді мінімізації функції втрат. Система машинного навчання обчислює функцію $y_i = f(x_i, w)$, де x_i – i -й вхідний вектор, w – параметри мережі в системі. Тут y_i являє собою розпізнаний клас для заданого вхідного вектора x_i – кількісну оцінку або показник ймовірності, пов'язаний з кожним класом. Якщо t_i – бажаний клас, N – кількість векторів, то функція втрати може бути представлена у вигляді $E(w)$, где w – вектор ваги або параметри мережі.

Часто вживаною функцією втрат для завдання регресії є середньоквадратична помилка (MSE - Mean Squared Error) [15], яка визначається як

$$E(w) = \frac{1}{N} \sum_{i=1}^N ||t_i - y_i||^2 \quad (1.1)$$

Загальноприйнятою функцією втрат для завдання класифікації є крос-ентропія [15], яка визначається як

$$E(w) = \frac{1}{N} \sum_{i=1}^N [t_i \log y_i + (1 - t_i) \log(1 - y_i)] \quad (1.2)$$

Оптимізація функції втрат - це процес пошуку таких вагових параметрів w , які дозволять мінімізувати $E(w)$. Для обчислення параметрів, які мінімізують функцію втрат, виконується ряд ітерацій.

Висновки до розділу 2

У цьому розділі даються теоретичні основи, необхідні для розуміння методів, що будуть використовуватись в наступних розділах. По-перше, обговорюються фундаментальні поняття машинного і глибокого навчання. По-друге, проводиться порівняльний аналіз машинного та глибокого навчання, що дає нам можливість досягнути всі недоліки та переваги поданих шляхів для навчання системи.

| | | | | | | |
|------|------|----------|--------|------|--------------------|------|
| | | | | | IT51.200БАК.007 Д5 | Лист |
| Ізм. | Лист | № докум. | Підпис | Дата | | |

3. АНАЛІЗ ЗАСОБІВ РЕАЛІЗАЦІЇ

3.1 Обґрунтування вибору типу моделі

За останнє десятиліття була пророблена певна робота по автоматизації процесу оцінки цін на нерухомість. Успіхи полягали в тому, щоб підкреслити атрибути об'єкта, такі як ділянка об'єкта, якість об'єкта, навколишнє середовище і місце розташування. Порівнюючи різні методи, ми виявили, що попередні підходи можна розділити на дві основні категорії: моделі на основі дезагрегації даних і моделі на основі агрегації даних. Моделі, засновані на дезагрегації, намагаються передбачити ціну будинку враховуючи кожен окремий атрибут, як приклад, гедонічна теорія цін. Однак моделі агрегації даних залежать від всіх атрибутів будинку в сукупності для оцінки його вартості, таких як нейронна мережа та регресивні моделі. Як приклад моделей дезагрегації даних можна привести гедоністичну теорію цін, в якій ціна нерухомості є функцією її атрибутів. Атрибути, пов'язані з нерухомістю, визначають набір неявних цін. Граничні неявні значення атрибутів отримані шляхом диференціювання гедонічним функції ціни по кожному атрибуту. Проблема цього методу полягає в тому, що він не враховує відмінності між різними властивостями в одному і тому ж географічному районі. Ось чому ця модель вважається необ'єктивною. Flitcher et al. (Fletcher et al., 2000) намагалися знайти найкращий спосіб оцінки вартості нерухомості, порівнюючи результати агрегації і дезагрегації даних. Вони виявили, що результати агрегації є більш точними. Вони також виявили, що гедоністичних ціна деяких коефіцієнтів для деяких атрибутів нестабільна, так як вони змінюються в залежності від місця розташування, віку і типу майна. Тому вони зрозуміли, що гедоністичний аналіз може бути ефективним при аналізі цих змін, але не для оцінки ціни на основі кожного атрибута окремо. Крім того, вони виявили, що географічне положення майна грає важливу роль в вплив на ціну майна. Для моделі агрегування

| | | | | | | |
|------|------|----------|--------|------|--------------------|------|
| | | | | | IT51.200БАК.007 Д5 | Лист |
| Ізм. | Лист | № докум. | Підпис | Дата | | |

даних нейронна мережа найпоширеніша модель. Бін Хаміс в (Khamis and Kamarudin, 2014 року) порівняв продуктивність нейронної мережі проти багатолінійної регресії (MLR). NN досяг вищого значення R^2 і нижчого MSE, ніж MLR. Порівнюючи результати гедонічної моделі з моделлю нейронної мережі, нейронна мережа перевершує гедонічну модель, досягаючи більш високого значення R^2 на 45,348% і нижчого MSE на 48,8441%. Недолік інформації в гедоністичній моделі може бути причиною поганої роботи. Однак в моделі нейронної мережі є деякі обмеження, оскільки передбачувана ціна не є реальною ціною, але близька до реальної. Це пов'язано з труднощами отримання реальних даних з ринку. Крім того, ефект часу грає важливу роль в процесі оцінки, який нейронні мережі не можуть обробляти автоматично. Це має на увазі, що на вартість нерухомості впливають багато інших економічних чинників, які важко включити в процес оцінки. Саме тому мною було обрано модель нейронної мережі з агрегацією даних.

3.2 Опис набору даних

Зібраний набір даних складається з 535 вибіркового будинків зі штату Каліфорнія в Сполучених Штатах. Кожен будинок представлений як візуальними, так і текстовими даними.

Візуальні дані представляють собою набір з 4 зображень для фронтального вигляду будинку, спальні, кухні і ванної кімнати, як показано на рисунку 3.1. Текстові дані представляють фізичні атрибути будинку, такі як кількість спалень, кількість ванних кімнат, площа будинку і поштовий індекс місця, де знаходиться будинок. Цей набір даних був зібраний і анотований вручну із загальнодоступної інформації на сайтах, які продають будинки. Немає ні повторних даних, ні пропущених. Ціна будинку в наборі даних коливається від 22 000 до 5 858 000 доларів. Таблиця 3.1 містить деякі

| | | | | | | |
|------|------|----------|--------|------|--------------------|------|
| | | | | | IT51.200БАК.007 Д5 | Лист |
| Ізм. | Лист | № докум. | Підпис | Дата | | |

статистичні дані про набір даних. Цей набір даних загальнодоступний для подальшого дослідження.



Рисунок 3.1 – Приклад зображень будинків з вибірки даних

Таблиця 3.1 – Таблиця характеристик будинків з вибірки даних

| Характеристика | Малий | Середній | Максимальн ий |
|----------------------------|----------|-----------|------------------|
| Ціна будинку (USD) | \$22,000 | \$589,360 | \$5,858,000 |
| Площа будинку (кв. м) | 701 | 2364.9 | 9583 |
| Кількість спалень | 1 | 3.38 | 10 |
| Кількість ванних кімнат | 1 | 2.67 | 7 |
| Розмір зображення | 250x187 | 801x560 | 1484x1484 |

3.3 Фільтр анізотропної функції

Фільтр анізотропної дифузії, запропонований Перон і Маліком [30], спрямований на зниження шуму на цифровому зображенні. При цьому виключено видалення значимої інформації, такої як краї, лінії і інші деталі зображення. Анізотропна дифузія є одним з ранніх методів напрямлених на

зменшення шуму з використанням приватних похідних. Вона дозволяє успішно згладжувати текстуру зображення, не розмиваючи кордони і невеликі структури всередині самого зображення. Це дозволяє спростити сегментацію, оскільки несуттєвих деталей на зображенні стає менше. На відміну від звичайних згладжуючих фільтрів анізотропна дифузія схожа на адаптивні методи, які згладжують зображення всередині області, залишаючи межі недоторканими. Фільтр використовувався в різних додатках комп'ютерного зору і дозволив поліпшити результати згладжування [31].

Нехай (x, y) - координата пікселя. Якщо (x, y) є частиною граничного пікселя, то анізотропна дифузія буде застосовувати низьке згладжування, в іншому випадку - висока. Основне рівняння дифузії має такий вигляд:

$$\frac{\partial I(x, y, t)}{\partial t} = \text{div}[g(|\nabla I(x, y, t)|)]\nabla I(x, y, t) \quad (3.1)$$

де t - параметр розмиття,

$I(x, y, t)$ - вхідне зображення,

div - оператор дивергенції,

g - функція провідності,

$\nabla I(x, y, t)$ - градієнт.

Функція g вибирається таким чином, щоб задовольнити рівності

$$\lim_{x \rightarrow 0} g(x) = 1 \quad (3.2)$$

при якому дифузія досягає максимуму всередині області, і

$$\lim_{x \rightarrow \infty} g(x) = 0 \quad (3.3)$$

при якому дифузія мінімальна на кордонах зображення. Перона і Малик запропонували дві функції провідності [30]:

$$g_1(x) = e^{-\left(\frac{x}{K}\right)^2} \quad (3.4)$$

і

$$g_2(x) = \frac{1}{1 + \left(\frac{x}{K}\right)^2} \quad (3.5)$$

де K - параметр, який контролює швидкість дифузії і служить м'яким порогом між зашумленими градієнтами зображення і градієнтами кордонів.

| | | | | | | |
|------|------|----------|--------|------|--------------------|------|
| | | | | | IT51.200БАК.007 Д5 | Лист |
| Ізм. | Лист | № докум. | Підпис | Дата | | |

Дискретна функція анізотропної дифузії задана як:

$$I_{t+1}(s) = I_t(s) + \frac{\gamma}{\beta_s} \sum_{p \in \beta_s} g_k(|\nabla I_{s,p}|) \nabla I_{s,p} \quad (3.6)$$

де I - дискретизованої зображення,

s - стан пікселя в двовимірної дискретної сітці,

$\gamma \in [0,1]$,

β_s є чотирехсвязную область для пікселя s (напрямок пікселя).

Функції провідності для фільтра анізотропної дифузії, запропоновані Маліком і Перон, працюють по-різному. Функція g_1 сприяє кордонів більш високої контрастності, а g_2 - широким областям зображення. Низьке значення порога K дозволяє отримати невеликі значення градієнтів яскравості в кожній точці і блокувати провідність і дифузію на кордонах.

Висновки до розділу 3

Враховуючи всю інформацію, наведену вище, можна зробити висновок, що під час розробки програмного продукту будуть враховані всі нюанси для формування раціональної моделі, яка забезпечить користувача об'єктивними та доволі точними даними. Розділ 3 дає розуміння того, чому було обрано саме наведені вище технології серед неймовірної кількості аналогів.

| | | | | | | |
|------|------|----------|--------|------|--------------------|------|
| | | | | | ІТ51.200БАК.007 Д5 | Лист |
| Ізм. | Лист | № докум. | Підпис | Дата | | |

4. ОСНОВНА СИСТЕМА

4.1 Опис основної системи

Основна мета нашого дослідження - перевірити вплив включення візуальних характеристик будинків, які будуть використовуватися для їх оцінки. Крім того, я спробував знайти взаємозв'язок між числовими візуальними характеристиками і точністю оцінки з використанням регресії опорних векторів і моделі нейронних мереж. Як показано на малюнку 4.1, наша система має різні етапи обробки, кожен з яких представлений модульним блоком.



Рисунок 4.1 – Пропонована система обробки

Перший модуль в нашій системі - обробка зображень де метод вирівнювання гістограми використовується для збільшення загального контрасту зображень набору даних. Цей метод привів до кращого розподілу інтенсивності кольору серед всіх зображень і дозволив областям з більш низьким локальним контрастом отримати високий контраст, ефективно поширюючи найбільш часті значення інтенсивності. Після цього екстрактор прискореної роботи (SURF) використовується для вилучення візуальних особливостей із зображень. SURF використовує цілочисельну апроксимацію детермінанта детектора крапель Гессен, яка може бути обчислена за допомогою 3 цілочисельних операцій з використанням попередньо обчисленого інтегрального зображення. Його характеристичний дескриптор заснований на сумі вейвлет-відгуку Хаара навколо точки, що нас цікавить. Вони також можуть бути розраховані за допомогою інтегрального зображення. На цьому етапі з кожного зображення будинку були вилучені найсильніші n об'єктів,

потім ці об'єкти були об'єднані разом у векторному форматі разом з текстовими атрибутами того ж будинку в певному порядку, щоб представити загальні характеристики цього будинку. Малюнок 4.2 є прикладом для витягнутого SURF особливості з 4 зображень будинку в наборі даних. Виділені елементи підкреслюють кути, різкі переходи і краю. Візуально було встановлено, що ці особливості зазначають пункти інтересів на зображеннях, як показано на фронтальному зображенні будинку, де вікна були обрані в якості важливих функцій.



Рисунок 4.2 – Приклад виокремлення SURF ознак з набору даних

Значення для витягнутих ознак n варіювалося від одного експерименту до іншого, як буде пояснено в розділі 5. Екстрактор ознак SURF дав кращі результати в порівнянні з екстрактором Scale Invariant Feature Transform (SIFT), і тому він також був швидше, це використовувалося у всіх наших експериментах. В останньому модулі агреговані функції передаються одному з модулів оцінки: SVR або NN після нормалізації. Нормалізація - це метод

попередньої обробки, коли дані масштабуються в діапазоні від 0 до 1. Формула, яка використовується для нормалізації:

$$z_i = \frac{x_i - \min(x)}{\max(x) - \min(x)} \quad (4.1)$$

4.2 Підтримка векторної регресії (SVR)

Мащини опорних векторів - це підходи машинного навчання для вирішення завдань оцінки багатовимірних функцій і регресії. SVM засновані на статистичній теорії навчання і засновані на принципі оптимального поділу класів. SVM використовують простір просторових об'єктів для вивчення і отримання функцій прогнозування, які розширюються на підмножині опорних векторів. Існують дві основні категорії для SVM: класифікація опорних векторів (SVC) і регресія опорних векторів (SVR). У SVC SVM намагаються розділити класи з мінімальною помилкою узагальнення, якщо класи розділяються. Якщо класи не є розділяються, SVM намагаються отримати гіперплощину, яка максимізує запас і зменшує помилку помилкової класифікації. У SVR Вапник ввів альтернативну функцію інтенсивних втрат, яка дозволяє використовувати запас для регресії. Основна мета SVR полягає в тому, щоб знайти функцію $f(x)$, яка має найбільшу ϵ відхилення від фактично одержаних цілей для всіх навчальних даних і в той же час як можна більш плоска. Іншими словами, помилка навчальних даних повинна бути менше ϵ , тому SVR залежить тільки від підмножини навчальних даних, тому що функція вартості ігнорує будь-які навчальні дані, які близькі або знаходяться в межах ϵ до прогнозу моделі. Це також вказує на те, що SVR вимагає ретельного вибору типу функції ядра і параметра регуляризації. Функція ядра може ефективно виконувати нелінійну регресію, неявно відображаючи вхідні дані в простір просторових об'єктів з більш високою розмірністю, щоб зробити

| | | | | | | |
|------|------|----------|--------|------|--------------------|------|
| | | | | | IT51.200БАК.007 Д5 | Лист |
| Ізм. | Лист | № докум. | Підпис | Дата | | |

можливим виконання лінійної регресії. Параметр визначає компроміс між площинністю функції і величиною, з якої допускаються відхилення від помилки, що перевищують ε . У наших експериментах ядро перетину гістограми було обрано в якості типу ядра, і оптимальне значення параметра (C) було отримано після декількох експериментів з набором даних для отримання найкращого результату. Перетин гістограми - це метод, запропонований в для індексації кольору із застосуванням розпізнавання об'єктів, і було доведено, що його можна використовувати в якості ядра для SVM як ефективного представлення систем розпізнавання на основі кольору, стійких до оклюзії і зміни зору. Метриками для оцінки ефективності SVR є коефіцієнт детермінації (R^2) і середня квадратична помилка (MSE).

4.3 Оцінка ефективності

4.3.1 Помилка середньої площі

Середньоквадратична помилка - це показник наближення оцінки до фактичних даних. Він вимірює середнє значення квадрата відхилень помилок розрахункових значень щодо фактичних значень. Він вимірюється:

$$MSE = \frac{1}{n} \sum_{i=1}^n (\hat{y} - y)^2 \quad (4.2)$$

де \hat{y} - розрахункова величина від регресії і y - фактичне значення. Чим нижче MSE, тим краща модель оцінки.

4.3.2 Коефіцієнт визначення

| | | | | | | |
|------|------|----------|--------|------|--------------------|------|
| | | | | | IT51.200BAK.007 Д5 | Лист |
| Ізм. | Лист | № докум. | Підпис | Дата | | |

Коефіцієнт детермінації є мірою близькості передбачуваної моделі щодо фактичної моделі. Обчислюється набір різних помилок:

$$SSE = \sum_{i=1}^n (\hat{y}_i - y_i)^2 \quad (4.3)$$

$$SST = \sum_{i=1}^n (\bar{y} - y_i)^2 \quad (4.4)$$

SSE - сума квадратів помилок, а SST - сума суми квадратів. Значення R-квадрат обчислюється за допомогою :

$$R^2 = 1 - \frac{SSE}{SST} \quad (4.5)$$

Значення R² знаходиться в діапазоні від 0 до 1, чим вище значення, тим точніше модель оцінки.

Висновки до розділу 4

У даному розділі я розглянув основну модель системи для розпізнавання елементів архітектури. Визначив основні помилки та шляхи їх вирішення. Розповів про основну концепцію роботи нейронної мережі та її особливості в області розпізнавання елементів на зображенні.

5. ЕКСПЕРЕМЕНТИ ТА РЕЗУЛЬТАТИ

5.1 SVR експерименти

У моделі SVR 428 будинків були використані для навчання, що становить 80% набору даних, і 107 будинків були використані для тестування, що становить 20% набору даних. SVR навчався і тестувався на різній кількості витягнутих ознак SURF кожен раз, щоб знайти взаємозв'язок між кількістю ознак і точністю моделі оцінки. Було протестовано 16 різних випадків, починаючи з навчання і тестування тільки з текстовими атрибутами без візуальних особливостей і просуваючись до вилучення додаткових функцій SURF до 15. У наших експериментах ядро перетину гістограми було обрано в якості типу ядра і оптимального значення для параметр (C) був отриманий після декількох експериментів з набором даних для отримання найкращого результату. Малюнки 6 і 7 в розділі 5.2 показують, що продуктивність SVR збільшується з додаванням більшої кількості візуальних функцій, поки він не досягне 9 зорових функцій, де модель досягає найменшого значення MSE 0,0037665 і найбільшого значення R 0,78602. Потім продуктивність SVR почала поступово погіршуватися після досягнення найвищої точки в 9 функціях.

5.2 Експеримент з нейронною мережею

Як показано на малюнку 4, ми прийняли повністю підключену архітектуру з одним прихованим шаром з 4 блоків. Очікувалося, що проблема буде нелінійною, тому в мережах є приховані шари. Кількість прихованих вузлів було вибрано так, щоб бути десь між числом вхідних і вихідних вузлів, і, спробувавши різну кількість нейронів в прихованому шарі, було доведено,

| | | | | | | |
|------|------|----------|--------|------|--------------------|------|
| | | | | | IT51.200БАК.007 Д5 | Лист |
| Ізм. | Лист | № докум. | Підпис | Дата | | |

що наявність 4 нейронів є оптимальною архітектурою. Наші нейрони мали функцію активації сигмовидної кишки і навчалися за допомогою варіації Левенберга Марквардта техніки зворотного поширення помилки. Ця архітектура дала кращі результати під час наших експериментів. Ми розділили наш набір даних на три частини: 70% для навчання, 15% для перевірки і 15% для тестування. Щоб уникнути перебору, ми припинили навчання після 5 епох, міра кількості раз, коли всі тренувальні вектори використовуються один раз для поновлення ваг, тому що помилка перевірки почала збільшуватися. На малюнку 5 показана продуктивність Мережі з виділенням навчальних, перевірочних та тестових MSE, а також коли процес навчання був зупинений щоб уникнути переоснащення. Малюнки 6 і 7 показують, що об'єднання 4 функцій SURF з текстовими атрибутами призводить до досягнення максимального значення R 0,95053 і найменшого MSE 0,000959. У моделі NN MSE починається дуже високо без візуальних особливостей, і зі збільшенням візуальних особливостей MSE починає зменшуватися, поки не досягне свого мінімального значення при 4 ознаках, а потім поступово починає збільшуватися до 16. Малюнок 6 показує, що NN перевершує модель SVR, досягнувши більш низького MSE на 76,02%. Крім того, на фіг.7 показано, що NN досяг вищого значення R на 21,05%, ніж SVR. Також на малюнку 8 показано, що лінія регресії, створювана NN, є більш точною, оскільки оцінні значення набагато ближче до фактичних даних.

5.3 Нейронна мережа з набору даних будинків

Щоб виключити залежність від даних, ми протестували нашу модель на наборі еталонних житлових даних (Lichman, 2013), який має 506 будинків, кожен з 13 текстовими атрибутами, такими як середня кількість кімнат в будинку, вік будинку, податок на вартість нерухомості і т. д. Ми порівняли наші результати з моделлю (Khamis and Kamarudin, 2014 року), яка використовувала NN для

| | | | | | | |
|------|------|----------|--------|------|--------------------|------|
| | | | | | IT51.200БАК.007 Д5 | Лист |
| Ізм. | Лист | № докум. | Підпис | Дата | | |

оцінки вартості будинку, ґрунтуючись тільки на текстових особливості. Ми відтворили їх модель, щоб мати можливість порівнювати результати як навчання, так і тестування, а не тільки навчання, про який повідомлялося в документі. Ми порівняли MSE і R-Value як в навчанні, так і в тестуванні, і наша модель перевершує їх модель. Середні ціни використовувалися при розрахунку MSE для порівняння результатів з моделлю (Lichman, 2013), тому значення MSE перевищують значення, вказані в нашому наборі даних. Результати, наведені в таблиці 2, показують, що наша модель досягає MSE $9,708 \times 10^6$ і R-значення 0,9348 на тестовому наборі, що краще, ніж модель Бін Хаміса, яка досягає MSE $1,713 \times 10^9$ і R-Value 0,87392. Наша модель досягає більш низької MSE на тренувальному наборі на 99,54% і на тестовому наборі на 99,43%. Він також досягає більш високого R-Value на 6,8% на тренувальному наборі і на тестовому наборі на 6,97%. Ці результати показують, що наша модель нейронної мережі не залежить від нашого власного набору даних.

1.1.1 Слой активации

Найбільш часто використовуваними функціями активації є сигмоїда гіперболічний тангенс, ReLU, ReLU з «витоком» [47].

Під сигмоїдальною функцією активації часто мається на увазі логістична функція яка визначається так:

$$f(x) = 1 / (1 + e^{(-x)}) \quad (4.8)$$

де x - деякий дійсне число.

Функція повертає число від 0 до 1. Дуже великі негативні числа на виході стають нулем, а дуже великі позитивні - одиницею. Сигмоїдальна функція, таким чином, дозволяє посилювати слабкі вхідні сигнали і при цьому не наситилися з сильних. При виборі сигмоид як функції активації при великих значеннях вихідних ваг мережа може погано вчитися, оскільки більшість нейронів перейде в стан насичення. Графік сигмоїдальної функції наведено на малюнку 4.3.

| | | | | | | |
|------|------|----------|--------|------|--------------------|------|
| | | | | | IT51.200БАК.007 Д5 | Лист |
| Ізм. | Лист | № докум. | Підпис | Дата | | |

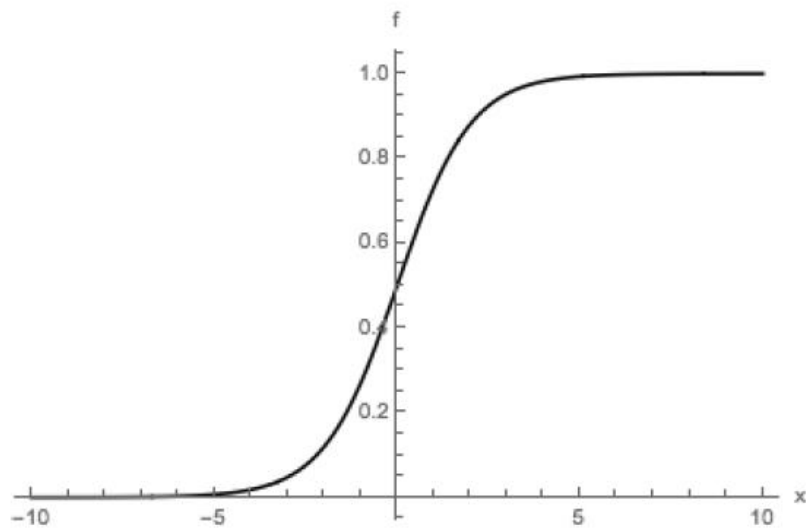


Рисунок 0.1. График сигмоидальной функции

Незважаючи на те, що логістична сигмоїда має добру біологічну інтерпретацію, кращим вибором в якості функції активації на практиці виявляється гіперболічний тангенс [48]:

$$\tanh(x) = \frac{2}{1+e^{-2x}} - 1 \quad (4.9)$$

где x – некоторое вещественное число.

Ця функція також є сигмоидальною («s»-образною), але її вихідні значення лежать в діапазоні $(-1, 1)$, причому тільки для нульових вхідних значень вихідні лежать в

околиці нуля, що дозволяє уникнути проблеми насичення на етапі навчання мережі. Графік функції гіперболічного тангенса наведено на малюнку 4.4.

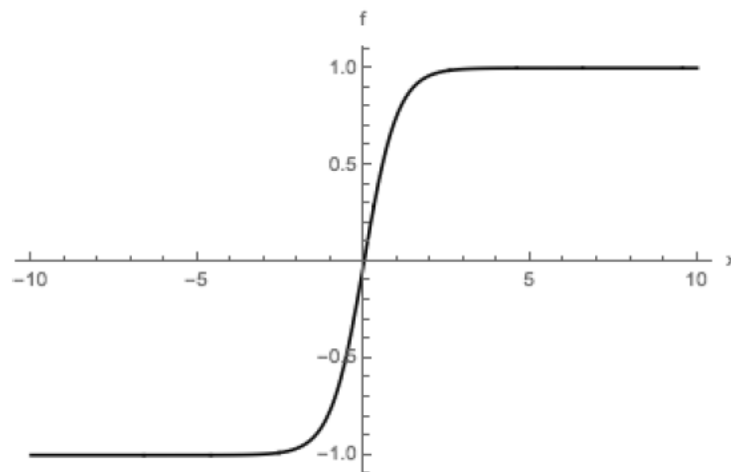


Рисунок 0.2. Графік функции гиперболического тангенса

В останні роки найбільшу популярність придбала функція активації ReLU («випрямляч») [49]:

$$f(x) = \max(0, x) \quad (4.10)$$

Функція ReLU не схильна до насичення. Вона реалізує пороговий перехід в нулі, не вимагає великих ресурсів, на відміну від сигмоидальної функції і гіперболічного тангенса, що виконують операцію піднесення до степеня. Це дозволяє прискорити процес навчання. На малюнку 4.5 проілюстровано графік функції ReLU.

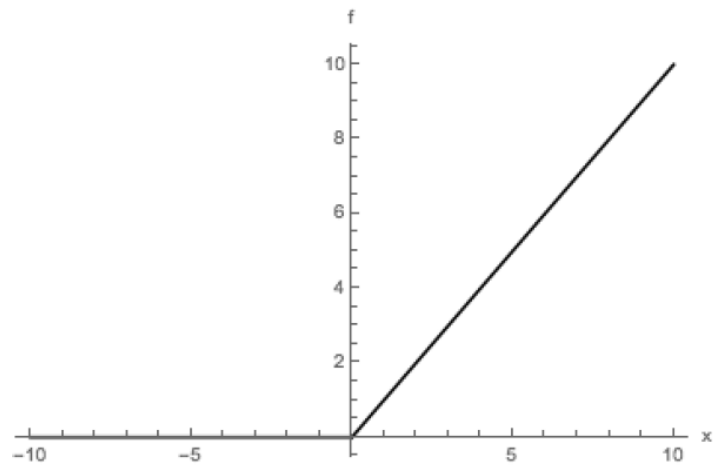


Рисунок 0.3. График функции ReLU

ReLU і її різні модифікації (ReLU «з витоком», параметричну ReLU, рандомізованих ReLU і інші) є найбільш часто використовуваними функціями активації в глибокому навчанні, зокрема, згортальних нейронних мережах [50].

1.1.2 Шар ПУЛІНГ

Шари ПУЛІНГ зменшують розмірність зображення і дозволяють використовувати меншу кількість параметрів, завдяки чому можна запобігти перенавчанню, оскільки після виявлення деяких ознак зображення ущільнюється до менш докладного. Крім того, пулінг забезпечує інваріантність до зрушень, спотворень і шуму. Операція ПУЛІНГ

здійснюється шляхом вертикального і / або горизонтального зсуву вікна (фільтра), як правило, розміру 2×2 по карті ознак. Крок ПУЛІНГ - це величина зміщення фільтра (в пікселях). Підвибірки - група пікселів зображення - ущільнюється до одного, проходячи деякий нелінійне перетворення (найчастіше вибирається середній або максимальне значення). Приклад роботи ПУЛІНГ наведено на малюнку 4.6.

1.1.2 Шар нормалізації

Нормалізація відгуку допомагає уповільнити швидкість навчання параметрів мережі [49]. Шар локальної нормалізації приносить користь при використанні нейронів з функцією активації ReLU - локальна околиця порушеної нейрона стає більш чутливою в порівнянні з її сусідами, однаково великі відгуки в будь-якій локальній околиці нейрона зменшуються, причому, якщо всі відгуки великі, нормалізація зменшує всі їх значення. Таким чином, цей шар є імітацією латерального гальмування - нейробіологічної концепції, яка описує здатність порушеної нейрона підпорядкувати своїх сусідів. Результат нормалізації $b_{-}(x, y)^i$ може бути формалізований наступним чином:

$$b_{-}(x, y)^i = (a_{-}(x, y)^i) / [(k + \alpha \sum_{j=\max\{0, i-n/2\}}^{\min\{N-1, i+n/2\}} [(a_{-}(x, y)^j)]^2)]^{\beta} \quad (4.13)$$

де $a_{-}(x, y)^i$ - активація нейрона, значення якого знайдено за допомогою фільтра i в позиції (x, y) ,

n - кількість сусідніх фільтрів в певному місці,

N - загальне число фільтрів в шарі,

k, α, β - деякі константи.

Значення n, k, α і β визначаються за допомогою валідаційної вибірки [49]. Використання шару нормалізації дозволяє зменшити тестову помилку: чотиришарова сверточное нейронна мережа на наборі даних CIFAR-10 досягла помилки 13% без нормалізації і 11% з нормалізацією [49].

повнозв'язну шар

Останній шар в архітектурі сверточное нейронної мережі є повнозв'язну, тобто сполученим зі всіма нейронами попереднього шару. Це прихований шар багат шарового

перцептрона, який діє як класифікатор - на виході виходить клас, до якого належить зображення. Нехай l - повнозв'язну шар, який складається з $m_l \wedge ((l-1))$ карт ознак. Тоді i -ий вихід $y_i \wedge ((l))$ шару l обчислюється за такою формулою:

$$y_i \wedge ((l)) = f(z_i \wedge ((l))), \text{ причому}$$

$$z_i \wedge ((l)) = \sum_{j=1}^{m_{l-1}} \left[\sum_{r=1}^{m_{l-2}} \left[\sum_{s=1}^{m_{l-3}} (W_{((i,j,r,s)) \wedge ((l))} - (u,v) (Y_j \wedge ((l-1)))) \right] \right] \quad (4.14)$$

де f - функція активації,

$W_{((i,j,r,s)) \wedge ((l))}$ - вагові коефіцієнти, що зв'язують нейрон в позиції (r, s) в карті ознак з номером j в шарі $(l-1)$ і нейрони карти ознак шару l з номером i .

Якщо попередній шар також є повнозв'язну, то його параметр $z_i \wedge ((l))$ описується як $z_i \wedge ((l)) = \sum_{k=1}^{m_{l-1}} [w_{((i,k)) \wedge ((l))} * Y_k \wedge ((l-1))]$, де $w_{((i,k)) \wedge ((l))}$ позначає вагу, що зв'язує i -ий нейрон шару $(l-1)$ з k -им елементом шару l .

Вихід шару є вектором, в якому кожен елемент являє собою передбачену мітку класу. Якщо вихідний шар мережі містить n класів, то цей вектор задається як $y = [y_1, y_2, \dots, y_n]^T$.

Softmax-шар

Особливий шар softmax може бути використаний в якості фінального шару мережі, слідом за останнім повнозв'язну шаром. Функція softmax є узагальненням логістичної функції активації на випадок многокласовой класифікації. Її значення трактуються як ймовірності приналежності об'єкта одному з класів. Функція перетворює вектор z розмірності K в вектор σ тієї ж розмірності. Кожна координата σ_i отриманого вектора представлена дійсним числом в інтервалі $[0,1]$, а сума координат дорівнює одиниці. Самі координати σ_i обчислюються так: $\sigma_{((z))} \wedge i = e^{(z_i)} / (\sum_{k=1}^K e^{(z_k)})$.

оптимізація продуктивності

Існує кілька методів, які дозволяють оптимізувати продуктивність сверточної нейронної мережі:

Поліпшення збіжності;

Боротьба з перенавчанням;

Обхід локальних мінімумів;

Підвищення точності або зменшення помилки розпізнавання.

Фактори, що впливають на продуктивність мережі, обговорюються нижче.

набори даних

Як правило, точність розпізнавання сверточное мережі залежить від набору даних, який повинен бути якомога більше [51]. Набір може бути розширено шляхом додавання декількох спотворених форм даних. Ці дані відіграють важливу роль в процесі навчання з використанням різних моделей мережі, дозволяючи забезпечити здатність мережі до узагальнення знайдених ознак. Як правило, всі доступні приклади діляться на навчальну і тестову вибірку в співвідношенні 80 до 20 [52].

момент

Момент, або імпульс, є методикою перерахунку ваг, яка дозволяє підвищити стабільність процесу навчання [52]. Імпульс допомагає параметрам домогтися своїх оптимальних значень і тим самим поліпшити швидкість збіжності мережі. Тут використовується усереднене значення змін параметрів на попередніх ітераціях для обчислення значення параметра на поточній ітерації. Ваги оновлюються наступним чином:

$$w(n+1) = w(n) + \alpha \Delta w(n) + \mu \Delta w(n-1) \quad (4.15)$$

де $w(n+1)$ - значення ваги на $n+1$ ітерації,

$\Delta w(n)$ - коригуючий зміна на n ітерації,

α - коефіцієнт імпульсу, що лежить між 0 і 1,

μ - коефіцієнт швидкості навчання.

У разі значення коефіцієнта імпульсу, рівного нулю, момент вироджується в метод градієнтного спуску, причому історія обчислення коригувальних значень на попередніх ітераціях не впливає на коригуючий значення поточної.

| | | | | | | |
|------|------|----------|--------|------|--------------------|------|
| | | | | | IT51.200БАК.007 Д5 | Лист |
| Ізм. | Лист | № докум. | Підпис | Дата | | |

1.2 Кількість згортальних шарів і карт ознак

При використанні згортальних мереж в додатках комп'ютерного зору вхідним параметром є зображення, а кількість вихідних параметрів залежить від кількості класів або категорій. При цьому кількість шарів згортки залежить від складності набору даних. Як відомо, згорткові шари допомагають в отриманні ознак низьких рівнів ієрархії. На жаль, точність вилучення може зменшуватися з кожним рівнем, що компенсується тимчасовими витратами на навчання. В кінцевому підсумку кількість згортальних шарів залежить від програми, тому найкращим способом є збільшення кількості шарів до отримання результатів, задовільних для конкретного завдання.

Аналітичного способу визначення кількості карт ознак в кожному сверточному шарі немає. Найпростіший підхід передбачає подвоєння кількості карт ознак з кожним шаром і точно так само залежить від набору даних. Найбільш популярною стратегією є використання завідомо великої кількості карт функцій з їх подальшим видаленням в залежності від часу навчання, використання пам'яті і т.д. [54]

1.3 Методологія

Завдяки швидким темпам зростання числа досліджень в області глибокого навчання великому інтересу до машинного навчання та аналізу даних в даний час є безліч інструментів і бібліотек з відкритим вихідним кодом, що реалізують алгоритми глибокого навчання. Майже кожен широко використовується мова програмування або має для цього деяку бібліотеку, або, по крайній мере, якийсь доступний API.

У даній роботі сверточна нейронна мережа використовується в якості екстрактора ознак і класифікатора для розпізнавання символів англійського алфавіту і чисел. Реєстраційний номер і ідентифікатор регіону транспортного засобу розпізнаються окремими моделями сверточної мережі. Навчання має на увазі наявність великого набору даних з прикладами реєстраційних номерів різних регіонів. Моделі згортальних нейронних мереж реалізуються з використанням бібліотек Caffe, OpenCV, Numpy, Scipy. Caffe - фреймворк для глибокого навчання, розроблений навчальним центром Berkeley Vision [55]. Він націлений в першу чергу на вирішення завдань класифікації та сегментації зображень, має одне з найбільших угруповань серед подібних інструментів для глибокого навчання і відрізняється швидкістю обчислень. Бібліотека OpenCV

| | | | | | | |
|------|------|----------|--------|------|--------------------|------|
| | | | | | IT51.200БАК.007 Д5 | Лист |
| Ізм. | Лист | № докум. | Підпис | Дата | | |

містить велику кількість алгоритмів комп'ютерного зору і обробки зображень. Модуль NumPy дозволяє ефективно працювати з тензорами (багатовимірними числовими масивами), а його комбінування з бібліотекою Scipy, призначеної для наукових розрахунків, надає багатий набір інструментів для обробки даних, в тому числі зображень. В якості мови програмування, призначеного для реалізації, був обраний високорівнева мова загального призначення Python. Вибір мови Python був обумовлений його популярністю в науковому співтоваристві, а також наявністю великої кількості бібліотек і розширень з відкритим вихідним кодом для роботи в області машинного навчання. Версія мови - 3.5.2, операційна система - Linux.

Навчання нейронної мережі вимагає великих обчислювальних витрат. З точки зору низкоуровневого програмування цей процес являє собою безліч матричних уможовин. Сучасні центральні процесори (CPU) не оптимізовані для подібних обчислень, в той час як GPU призначені для забезпечення точності таких операцій. Щоб виконати прискорення обчислень, потрібно високошвидкісний доступ до пам'яті. В даний час на ринку існують дві основні паралельні обчислювальні платформи - CUDA від NVIDIA і OpenCL, підтримуваний AMD проект з відкритим вихідним кодом. У даній роботі для навчання моделі сверточное нейронної мережі був обраний GPU від NVIDIA, оскільки CUDA на даний момент є лідером в області глибокого навчання. В якості моделі GPU був обраний графічний процесор GeForce GTX 980. Його специфікація включає в себе 2048 ядер CUDA, 256-бітну шину пам'яті, 4 Гб пам'яті GDDR5 з тактовою частотою 7 Гб / с, частоту 1216 МГц. В якості операційної системи обрана Fedora 21, що працює на 12-ядерному процесорі i7-5860. Тактова частота процесора становить 3 ГГц, є 64 Гб оперативної пам'яті і дві графічні карти GTX 980. В цілому навчання займає від 24 до 120 годин в залежності від архітектури сверточное нейронної мережі, яка обговорюється в наступному підрозділі.

1.3.1 Розпізнавання об'єктів

Символи, отримані в результаті сегментації, подаються на вхід сверточное мережі для розпізнавання. У цій архітектурі цифра «0», а також символи «Q» і «O» вважаються що належать до одного і того ж класу, тому кількість вихідних класів становить 34. Набір даних складається з зображень символів алфавіту і чисел. В результаті експериментів

| | | | | | | |
|------|------|----------|--------|------|--------------------|------|
| | | | | | IT51.200БАК.007 Д5 | Лист |
| Ізм. | Лист | № докум. | Підпис | Дата | | |

був обраний розмір 13000 зображень для навчальної вибірки і 3000 зображень для тестової. Всі зображення були нормалізовані і уніфіковані; кожне з них має розмір 32×32 пікселів і знаходиться в довічному форматі. У таблиці 4.1 коротко представлена інформація про кожного шарі сверточное мережі. Запис Conv + ReLU означає, що карти ознак, отримані на шарі Conv (скорочення від «convolutional» - рус. «Сверточних»), передаються через шар ReLU. Те ж саме вірно і для Pool + Norm (комбінація шарів ПУЛІНГ - pooling - і нормалізації - normalization). Результат, отриманий на повнозв'язну шарі (ip1), передається шару ReLU і проходить через шар дропаутов. Ще один повнозв'язну шар (ip2) відповідає за результат класифікації зображення.

Таблиця 4.1

Схема архітектури сверточное нейронної мережі для розпізнавання реєстраційних номерів з використанням двійкових зображень

| Слой | Входной размер (высота × ширина) | Размер фильтра (высота × ширина) | Число фильтров | Дополнение нулями (высота × ширина) | Шаг свертки |
|---------------------------------------|-------------------------------------|-------------------------------------|----------------|--|-------------|
| Conv1 | 32×32 | 3×5 | 60 | 1×1 | 1 |
| Conv2 + ReLU2 | 32×30 | 5×7 | 120 | 2×3 | 1 |
| Pool2 + Norm2 | 32×30 | 8×8 | 120 | - | 2 |
| Conv3 + ReLU3 | 32×30 | 5×1 | 384 | 2×0 | 1 |
| Pool3 + Norm3 | 32×30 | 2×2 | 384 | - | 1 |
| Полносвязный (ip1) + ReLU4 + Dropout4 | 32×30 | - | 584 | - | - |
| Полносвязный (ip2) | 584 | - | 34 | - | - |
| Softmax | 34 | - | 34 | - | - |

Числа і символи алфавіту навчаються на двох різних архітектурах. Друга архітектура навчається з використанням зображень у відтінках сірого, кожне з яких має розмір 24×14 пікселів. Навчальна вибірка складається з 9324 зображень, тестова - з

2318. У цій архітектурі всього 36 вихідних класів - все символи англійського алфавіту цифри. Повна інформація про архітектуру приведена в таблиці 4.2.

Таблиця 4.2

Схема архітектури сверточное нейронної мережі для розпізнавання реєстраційних номерів з використанням відтінків сірого

| Слой | Входной размер (высота × ширина) | Размер фильтра (высота × ширина) | Число фильтров | Дополнение нулями (высота × ширина) | Шаг свертки |
|--|---|--|-------------------|---|----------------|
| Conv1 | 24 × 14 | 5 × 5 | 74 | 2 × 2 | 1 |
| Conv2 + ReLU2 | 24 × 14 | 5 × 3 | 150 | 2 × 1 | 1 |
| Pool2 + Norm2 | 24 × 14 | 4 × 4 | 150 | - | 1 |
| Conv3 | 24 × 14 | 3 × 1 | 384 | 2 × 0 | 1 |
| Conv4 + ReLU4 | 24 × 14 | 3 × 3 | 680 | 1 × 1 | 1 |
| Pool4 + Norm4 | 24 × 14 | 4 × 4 | 680 | - | 1 |
| Conv5 | 24 × 14 | 3 × 3 | 784 | 1 × 1 | 1 |
| Conv6 + ReLU6 | 24 × 14 | 3 × 3 | 784 | 1 × 1 | 1 |
| Полносвязный (ip1) + ReLU7 + Dropout7 | 24 × 14 | - | 500 | - | - |
| Полносвязный (ip2) | 500 | - | 36 | - | - |
| Softmax | 36 | - | 36 | - | - |

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Computer Vision [Эл. ресурс] / Режим доступа: https://en.wikipedia.org/wiki/Computer_vision (Дата обращения: 30.06.2018)
2. Large Scale Visual Recognition Challenge 2012 (ILSVRC2012) [Эл. ресурс]. / Режим доступа: <http://www.image-net.org/challenges/LSVRC/2012/results.html> (Дата обращения: 30.06.2018)
3. Baird, Henry S., and Karl Tombre. "The Evolution of Document Image Analysis." Handbook of Document Image Processing and Recognition. Springer London, 2014. 63-71.
4. Recognizing Text in Google Street View Images - UCSD CSE [Эл. ресурс] / Режим доступа: <https://cseweb.ucsd.edu/classes/wi10/cse190-a/reports/jlintern.pdf> (Дата обращения: 01.07.2018)
5. Haykin, S. "Neural Networks and Learning Machines". Pearson Education, 2011. 107-108.
6. Bengio Y. et al. "Deep learning of representations for unsupervised and transfer learning", ICML Unsupervised and Transfer Learning 27 (2012), 17–36.
7. Bengio, Y. "Learning deep architectures for ai," Found. Trends Mach. Learning, vol. 2, no. 1, pp. 1-127, Jan. 2009. [Эл. ресурс] / Режим доступа: <http://dx.doi.org/10.1561/22000000006> (Дата обращения: 02.07.2018)
8. C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," arXiv preprint arXiv:1409.4842, 2014.
9. R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in Comput. Vision and Pattern Recognition (CVPR), 2014 IEEE Conf., pp. 580-587.
10. Q. V. Le, "Building high-level features using large scale unsupervised learning," in Acoust., Speech and Signal Processing (ICASSP), 2013 IEEE Int. Conf., pp. 8595-8598.
11. Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," Proc. IEEE, vol. 86, no. 11, pp. 2278-2324, 1998.

12. G. E. Dahl, D. Yu, L. Deng, and A. Acero, "Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition," *Audio, Speech, and Language Processing, IEEE Trans. on*, vol. 20, no. 1, pp. 30-42, 2012.
13. D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. Journal of Computer Vision*, vol. 60, no. 2, pp. 91-110, 2004.
14. N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Comput. Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Comput. Soc. Conf.* vol. 1, pp. 886-893.
15. Y. LeCun and M. Ranzato, "Deep learning tutorial," in *Tutorials in Int. Conf. on Machine Learning (ICML'13)*, 2013.
16. Hyun Ah Song, Soo-Young Lee "Hierarchical Representation Using NMF", *ICONIP 2013: Neural Information Processing* pp. 466-473
17. G. E. Nasr, E. Badr, and C. Joun, "Cross entropy error function in neural networks Forecasting gasoline demand." in *FLAIRS Conference*, 2002, pp. 381-384.
18. Department of Electrical and Computer Web Pages. License Plate Recognition Final Report [Электронный ресурс] / Режим доступа: http://www.ece.cmu.edu/~ee551/Final_Reports/Gr18.551.S00.pdf (Дата обращения: 05.07.2018)
19. Инспектор+ – интеллектуальный охранный комплекс от компании ISS Technology [Электронный ресурс] / Режим доступа: http://www.sinf.ru/catalog/vi_iss/autoi.htm (Дата обращения: 05.05.2018)
20. Система распознавания автомобильных номеров "Auto-Trassir" [Электронный ресурс] / Режим доступа: <http://www.dssl.ru/products/autotrassir/> (Дата обращения: 05.07.2018)
21. Аппаратное распознавание номеров на базе IP-камер Hikvision и ПО TRASSIR [Электронный ресурс] / Режим доступа: <http://www.dssl.ru/company/articles/2017/apparatnoe-raspoznavanie-nomerov/> (Дата обращения: 05.07.2018)
22. АвтоУраган – Технологии распознавания [Электронный ресурс] / Режим доступа: <http://www.recognize.ru/ru/produktyi/avtouragan.html> (Дата обращения: 05.07.2018)

23. Новые Технологии :: CVS | Система распознавания автомобильных номеров [Электронный ресурс] / Режим доступа: http://www.cvsnt.ru/products_soft_cvsauto.html (Дата обращения: 06.07.2018)
24. CVS Авто -Программный комплекс [Электронный ресурс] / Режим доступа: <http://www.techportal.ru/material/?id=14112> (Дата обращения: 06.07.2018)
25. Распознавание автомобильных номеров: «Авто-Инспектор» - Лидер тестирования [Электронный ресурс] / Режим доступа: https://www.adron-perm.ru/test_leader/ (Дата обращения: 06.07.2018)
26. C.-N. E. Anagnostopoulos, I. E. Anagnostopoulos, I. D. Psoroulas, V. Loumos, and E. Kayafas, "License plate recognition from still images and video sequences: A survey," IEEE Trans. Intell. Transp. Syst., vol. 9, no. 3, pp. 377-391, 2008.
27. R. Prates, G. Camara-Chavez, W. R. Schwartz, and D. Menotti, "Brazilian license plate detection using histogram of oriented gradients and sliding windows," arXiv preprint arXiv:1401.1990, 2014.
28. Y.-N. Chen, C.-C. Han, C.-T. Wang, B.-S. Jeng, and K.-C. Fan, "The application of a convolution neural network on face and license plate detection," in Pattern Recognition, 2006. ICPR 2006. 18th Int. Conf., vol. 3. IEEE, 2006, pp. 552-555.
29. S. A. Radzi and M. Khalil-Hani, "Character recognition of license plate number using convolutional neural network," in Visual Informatics: Sustaining Research and Innovations. Berlin: Springer, 2011, pp. 45-55.
30. P. Perona and J. Malik, "Scale-space and edge detection using anisotropic diffusion," IEEE Trans. Pattern Anal. Mach. Intell., vol. 12, no. 7, pp. 629-639, 1990.
31. J. Weickert, "A review of nonlinear diffusion filtering," in Scale-space Theory in Computer Vision. Berlin: Springer, 1997, pp. 1-28.
32. R. Gonzalez and R. Woods, Digital Image Processing. Pearson Education, 2011
33. K. Zuiderveld, "Contrast limited adaptive histogram equalization," in Graphics gems IV. San Diego: Academic Press Professional, Inc., 1994, pp. 474-485.
34. Yang Y., Yan H. An adaptive logical method for binarization of degraded document images, Pattern Recognition, 2010, Vol. 33, pp. 787-807
35. J. Sauvola and M. Pietikainen, "Adaptive document image binarization," Pattern Recognition, vol. 33, no. 2, pp. 225-236, 2000.

36. Bar-Yosef, Itay and Hagbi, Nate and Kedem, Klara and Dinstein, Itshak. Line segmentation for degraded handwritten historical documents // Document Analysis and Recognition, 2009. ICDAR'09. 10th International Conference on, (1161–1165), 2009
37. S. Du, M. Ibrahim, M. Shehata, and W. Badawy, "Automatic license plate recognition (alpr): A state-of-the-art review," IEEE Trans. Circuits and Syst. for Video Technol. vol. 23, no. 2, pp. 311-325, 2013.
38. R. Kresch and D. Malah. Skeleton-based morphological coding of binary images. IEEE Transactions on Image Processing, IP-7(10):1387–1394, 1998.
39. T. Nukano and M. Khalid, "“vehicle license plate character recognition by neural networks,” pp. 771–775, in Proc. Int. Symp. Intell. Signal Process. Commun. Syst. 2004.
40. K. K. Kim, K. Kim, J. Kim, and H. J. Kim, "Learning-based approach for license plate recognition," pp. 614–623, Proceedings of the IEEE Signal Processing Society Workshop, Vol. 2, 2000.
41. X. Xi, H. Shi, L. Han, T. Wang, H. Y. Ding, G. Zhang, Y. Tang, and Y. Yin, "Breast tumor segmentation with prior knowledge learning," Neurocomputing, vol. 237, no. Supplement C, pp. 145 – 157, 2017
42. Bryan Catanzaro, Bor-Yiing Su, Narayanan Sundaram, Yunsup Lee, Mark Murphy, and Kurt Keutzer. "Efficient, high-quality image contour detection". ICCV, 2009
43. Skansi S. "Introduction to Deep Learning: From Logical Calculus to Artificial Intelligence Springer", 1st ed., 2018
44. Z. Zhao, S. Yang, and X. Ma, "Chinese license plate recognition using a convolutional neural network," in Computational Intell. and Ind. Application, 2008.PACIIA'08 Pacific-Asia Workshop on, vol. 1. IEEE, 2008, pp. 27-30.
45. Nielsen Michael A. "Neural Networks and Deep Learning" [Эл. ресурс]. / Режим доступа: <http://www.neuralnetworksanddeeplearning.com> (Дата обращения 03.09.2018)
46. Y. LeCun, K. Kavukcuoglu, and C. Farabet, "Convolutional networks and applications in vision," in Circuits and Syst. (ISCAS), Proc. of 2010 IEEE Int.Symp., pp. 253-256.

47. Karpathy A. et al. "Convolutional Neural Networks for Visual Recognition" [Эл. ресурс] / Режим доступа: <http://www.cs231n.stanford.edu> (Дата обращения: 11.05.2018)
48. Russell S. J., Norvig P. "Artificial Intelligence: A Modern Approach", 2nd ed. Prentice Hall, c2002. ISBN 978-0137903955.
49. A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in Advances Neural Inform. Proc. Syst., 2012, pp. 1097-1105.
50. Goodfellow I., Bengio Y., Courville A. "Deep Learning", MIT Press, 2016
51. R. O. Duda, P. E. Hart, and D. G. Stork, Pattern classification. New York: John Wiley & Sons, 2012.
52. C. Bishop, Pattern Recognition and Machine Learning, ser. Information Science and Statistics. New York: Springer, 2006. [Эл. ресурс] / Режим доступа: <https://books.google.com/books?id=kTNoQgAACAAJ> (Дата обращения: 01.11.2018)
53. Chatfield K., Simonyan K., Vedaldi A., Zisserman A. "Return of the devil in the details: Delving deep into convolutional nets", arXiv preprint arXiv: 1405.3531 (2014)
54. S. Theodoridis and K. Koutroumbas, Pattern Recognition. New York: Elsevier Science, 2008. [Эл. ресурс] / Режим доступа: <https://books.google.com/books?id=QgD-3Tcj8DkC> (Дата обращения: 05.11.2018)
55. Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," arXiv preprint arXiv:1408.5093, 2014.
56. P. Y. Simard, D. Steinkraus, and J. C. Platt, "Best practices for convolutional neural networks applied to visual document analysis," in Advances in Neural Information Processing Systems 5 (NIPS'92). IEEE, 1992, p. 958.
57. Y. Bengio, I. J. Goodfellow, and A. Courville, "Deep learning," 2015, book in preparation for MIT Press. [Эл. ресурс] / Режим доступа: <https://www.labs.iro.umontreal.ca/~bengioy/dlbook/> (Дата обращения: 12.11.2018)
58. P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, «Overfeat: Integrated recognition, localization and detection using convolutional networks», arXiv preprint arXiv:1312.6229, 2013.